REMARKS

STATUS OF THE CLAIMS

Claims 53-55 and 58 have been pending in the application.

Claims 53-55 and 58 are rejected under 35 USC 102(b) as being anticipated by Nelson, "Running Visual Basic for Windows," Microsoft Press, 1993.

Claims 53-55, and 58 are rejected under 35 U.S.C. 102(b) as being anticipated by Maxfield, "Designing With Objects", MachTech, Vol. 7, Issue No. 1, 1991.

According to the foregoing, the claims are amended, new claims 59-60 are added, and, thus, reconsideration of the pending claims, including the new claims, is respectfully requested.

No new matter has been added.

REJECTION

The Examiner maintains from the previous Office Action rejection of claims 53-55 and 58 under 35 USC 102(b) as being anticipated by Nelson, "Running Visual Basic for Windows," Microsoft Press, 1993 (hereinafter referred to as "Visual Basic").

The Examiner newly rejects claims 53-55 and 58 under 35 USC 102(b) as being anticipated by Maxfield, "Designing With Objects," MachTech, Vol. 7, Issue No. 1, 1991. Maxfield is newly cited, and, thus, newly relied upon.

Visual Basic and Maxfield fail to disclose or suggest the claimed present invention, because although they discuss object oriented programming object definition, they fail to disclose or suggest the claimed present invention's, "***object oriented programming ... component to serve as one object in combination with existing software ... by storing data related to the existing software ... and a program code ... issuing an event of the existing software in response to a received message issued in another object***" (e.g., previously presented independent claim 53).

However, taking into consideration the Office Action rejection rationale in the Response to Arguments, that Visual Basic provides "command" button creation as an object (Visual Basic, pages 9-13, page 2, lines 6-10 of item 2 of the Office Action), that "data and methods are elements of a program function/object" (page 2, lines 11-14 of item 2 of Office Action), and that "the claims are broad that can be read the component builder of Nelson" (page 3, lines 7-8 of the Office Action), the independent claims 53, 55 and 58, using claim 53 as an example, are amended to clarify the claimed present invention, as follows:

> 53. (CURRENTLY AMENDED) A computer readable storage ~~that stores~~medium storing an object oriented programming ~~objects~~component accessible by objects, comprising:
>
> ~~a data structure that defines a component to serve as one object in combination with existing software having a graphical user interface, by~~
>
> ~~storing~~ ***data related to ~~the~~ existing software having ~~the~~a graphical user interface (GUI),*** and
>
> ~~storing as an operation~~***as a component method, ~~related to the existing software,~~ a program code ~~controlling a data processing~~to control a computer system according to a process of driving the existing software, including issuing an event of the existing software, based upon the data related to the existing software, in response to a received message issued in another object to the existing software,***
>
> wherein ***the data and the component method together in combination with the existing software serve as one object***.

Support for the claim amendments can be found, for example, in page 178, line 17 to page 179, line 5 and FIG. 101 of the present Application.

Visual Basic's or Maxfield's objects do not comprise "***a program code** … **driving the existing software … in response to a received message*** issued in another object ***to the existing software***" as claimed. A conventional OOP class or object definition does not contain the claimed present invention's, "***data related to ~~the~~ existing software*** having ~~the~~a graphical user interface," and "a program code ~~controlling a data processing~~to control a computer system according to a process of ***driving the existing software, including issuing an event of the existing software, based upon the data related to the existing software, in response to a received message*** issued in another object ***to the existing software***," so that "***the data and the component method together in combination with the existing software serve as one object***."

In other words, Visual Basic and Maxfield, discuss describing the attributes of an object, whose settings can control the appearance and behavior of the object in the application (e.g., pages 6 and 9 of Visual Basic and objects as circles in diagrams of Maxfield). However, neither Visual Basic or Maxfield, disclose or suggest the claimed present invention’s, “***an object oriented programming ~~objects~~component*** … comprising: ***data related to ~~the~~ existing software***.” It is readily apparent that none of the object properties discussed by Visual Basic, for example, Figure 1-5 and 1-6, is a property that relates to “***existing software***.” Maxfield is silent on object property details.

It is readily apparent that Visual Basic fails to discuss an object that comprises the claimed present invention’s, “***program code*** … ***driving the existing software, including issuing an event of the existing software, based upon the data related to the existing software, in response to a received message*** issued in another object ***to the existing software***,” so that “***the data and the component method together in combination with the existing software serve as one object***,” because Visual Basic’s object do not comprise any “***data related to ~~the~~ existing software***.”

Regarding Maxfield, the Examiner in page 5, item 5, alleges that the description, “Fourth, a rectangular box is used to indicate an external event which causes some data to flow through the system in the form of messages,” is similar to the claimed present invention’s “***existing software***.” However, it is readily apparent that an external event differs from the claimed present invention’s, “***an object oriented programming ~~objects~~component*** … comprising: ***a program code*** … ***driving the existing software, including issuing an event of the existing software, based upon the data related to the existing software, in response to a received message*** issued in another object ***to the existing software***,” so that “***the data and the component method together in combination with the existing software serve as one object***.” In other words, for example, in Maxfield’s diagram 1, the rectangular box SFGetFile Dialog, is not part of an object but external to an object, so Maxfield fails to discuss “***an object oriented programming ~~objects~~component*** … comprising: ***a program code*** … ***driving the existing software, including issuing an event of the existing software, based upon the data related to the existing software, in response to a received message*** issued in another object ***to the existing software***.” Visual Basic’s or Maxfield’s objects do not comprise “***a program code*** … ***driving the existing software … in response to a received message*** issued in another object ***to the existing software***” as claimed.

Visual Basic and Maxfield cannot anticipate the claimed present invention, because they do not disclose or suggest each and every element of the claimed present invention. In view of the claim amendments and remarks, withdrawal of the rejection of pending claims and allowance of pending claims is respectfully requested.

NEW CLAIMS

Further, in contrast to the relied upon references, new dependent claim 59, provides:

> 59. (NEW) A computer readable storage medium according to claim 53, wherein the data related to the existing software comprises GUI system identification information of the existing software.

Further, in contrast to the relied upon references, a new independent claim 60 provides an alternative recitation of the claimed present invention that also patentably distinguishes over the relied upon references, as follows:

> 60. (NEW) An object oriented programming component definition, comprising:
>
> ***system information related to an existing software***; and
>
> as a component method, ***a program code to drive the existing software*** by controlling an apparatus according to a process comprising:
>
> ***based upon the system information of the existing software, issuing an event*** of the existing software, ***or issuing a message to inform another object*** of the event of the existing software, or both, ***in response to a received message issued by another object*** to the component method driving the existing software, ***or in response to a predetermined event issued to the existing software***, or both,
>
> wherein ***the existing software system information and the component method together in combination with the existing software serve as one object***.

New independent claim 60 is a "process" type claim by providing a patentably distinguishing process of defining an OOP component and is statutory subject matter under 35 USC 101. Support for the claimed present invention, including the new claim 60, can be found, for example, in pages 175 to 187 (e.g., page 178, lines 17-24, page 185, lines 1-4) and FIGS. 101 to 108 of the present Application.

The new claims are patentably distinguishing over the relied upon references based upon the same patentability rationale discussed above for pending independent claims 53, 55, and 58, and allowance of the new claims is respectfully requested.

CONCLUSION

There being no further outstanding objections or rejections, it is submitted that the application is in condition for allowance. An early action to that effect is courteously solicited.

Finally, if there are any formal matters remaining after this response, the Examiner is requested to telephone the undersigned to attend to these matters.

Respectfully submitted,
STAAS & HALSEY LLP

Date: March 8, 2006

By: ______________________
Mehdi D. Sheikerz
Registration No. 41,307

1201 New York Avenue, NW, 7th Floor
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501